IN THE CLAIMS

Please amend the claims as follows:

1. (Currently Amended) A functional random instruction testing (FRIT) method for testing a complex device, comprising:

converting a FRIT kernel into kernel test patterns and storing the kernel test patterns in a tester memory;

loading, at the tester, the kernel test patterns stored in the tester memory onto an on-board memory of a complex device under test (DUT);

executing[[, at]] software within the FRIT kernel on the complex device under test (DUT), to perform a re-generative functional test of the complex device under test (DUT) ~~by applying the kernel test patterns to the complex device under test (DUT)~~; and

comparing, at the tester, a test result of the re-generative functional test with an expected test result to check for manufacturing defects.

2. (Previously Presented) The method as claimed in claim 1, wherein said FRIT kernel includes a software built-in self-test engine (SBE) to execute the re-generative functional test of the complex device under test (DUT), and the expected test result obtained from computer modeling of the complex device under test (DUT).

3. (Original) The method as claimed in claim 2, wherein said expected test result is obtained from computer modeling of the complex device under test (DUT) or from a known good device.

4. (Previously Presented) The method as claimed in claim 2, wherein said software built-in self-test engine (SBE) of the FRIT kernel comprises:

a RIT generator including compact RIT machine code reside in the on-board memory of the complex device under test (DUT) for generating the re-generated functional test;

a test program execution module including test execution directives for providing an environment to store and run the re-generated functional test; and

a test result compaction module including compression machine code to compress test results of the re-generated functional test for storage in the on-board memory of the complex device under test (DUT).

5. (Original) The method as claimed in claim 4, wherein said test execution environment employs an exception handler for handling illegal conditions such as undesirable memory accesses, deadlock, shut-down, and infinite loops.

6. (Previously Presented) The method as claimed in claim 1, wherein said complex device under test (DUT) includes a microprocessor.

7. (Currently Amended) The method as claimed in claim 6, wherein, when ~~the kernel test patterns are applied to the microprocessor~~ software within the FRIT kernel is executed from the on-board memory, the microprocessor performs the following:

beginning a set-up for executing ~~the kernel test patterns~~ a software built-in self-test engine (SBE) within the FRIT kernel;

executing the ~~kernel test patterns~~ software built-in self-test engine (SBE) to generate a series of test sequences and associated data for respective test sequences;

running the test sequences, and at the end of the test sequences, obtaining the test results for storage in the on-board memory; and

dumping the test results ~~of the kernel test patterns to the tester~~, via an interface, for making a comparison with the expected test result to check for manufacturing defects.

8. (Currently Amended) The method as claimed in claim 7, wherein said FRIT kernel includes [[a]] the software built-in self-test engine (SBE) to execute the re-generative functional test of the complex device under test (DUT), and the expected test result obtained either from computer modeling of the complex device under test (DUT) or from a known good device.

9. (Currently Amended) The method as claimed in claim [[8]] 7, wherein said software built-in self-test engine (SBE) of the FRIT kernel is programmed to generate and execute one or more ("N") instruction sequences, each sequence being executed on one or more (M) data sets, where N and M represent an integer no less than "1" and are user-specified numbers used in generating the FRIT kernel ~~by an especially designed software tool~~.

10. (Original) The method as claimed in claim 9, wherein said software built-in self-test engine (SBE) of the FRIT kernel is further programmed to generate one or more signatures to provide a unique identification of the test result of each test sequence and indicate whether the test result of a particular test sequence is "good" or "bad".

11. (Currently Amended) A computer readable medium having instructions stored thereon ~~a functional random instruction test (FRIT) kernel~~ which, when executed by a system, cause the system to perform:

receiving [[the]] a functional random instruction test (FRIT) kernel [[in]] as kernel test patterns and storing the kernel test patterns in a tester memory;

loading the kernel test patterns stored in the tester memory onto an on-board memory of a complex device under test (DUT);

enabling execution[[, at]] of software within the FRIT kernel on the complex device under test (DUT), to perform a re-generative functional test of the complex device under test (DUT) ~~by applying the kernel test patterns to the complex device under test (DUT)~~; and

making a comparison between a test result of the re-generative functional test and an expected test result to check for manufacturing defects.

12. (Previously Presented) The computer readable medium as claimed in claim 11, wherein said FRIT kernel includes a software built-in self-test engine (SBE) to execute the re-generative functional test of the complex device under test (DUT), and the expected test result.

13. (Original) The computer readable medium as claimed in claim 12, wherein said expected test result is obtained from computer modeling of the complex device under test (DUT) or from a known good device.

14. (Previously Presented) The computer readable medium as claimed in claim 12, wherein said software built-in self-test engine (SBE) of the FRIT kernel comprises:

a RIT generator including compact RIT machine code in the on-board memory of the complex device under test (DUT) for generating the re-generated functional test;

a test program execution module including test execution directives for providing an environment to store and run the re-generated functional test; and

a test result compaction module including compression machine code to compress test results of the re-generated functional test for storage in the on-board memory of the complex device under test (DUT).

15. (Original) The computer readable medium as claimed in claim 14, wherein said test execution environment employs an exception handler for handling illegal conditions such as undesirable memory accesses, deadlock, shut-down, and infinite loops.

16. (Previously Presented) The computer readable medium as claimed in claim 11, wherein said complex device under test (DUT) includes a microprocessor.

17. (Currently Amended) The computer readable medium as claimed in claim 16, wherein, when ~~the kernel test patterns are applied to the microprocessor~~ software within the FRIT kernel is executed from the on-board memory, the microprocessor performs the following:

beginning a set-up for executing ~~the kernel test patterns~~ a software built-in self-test engine (SBE) within the FRIT kernel;

executing the ~~kernel test patterns~~ software built-in self-test engine (SBE) to generate a series of test sequences and associated data for respective test sequences;

running the test sequences, and at the end of the test sequences, obtaining the test results for storage in the on-board memory; and

dumping the test results ~~of the kernel test patterns~~ to the testing system, via an interface, for making said comparison with the expected test result to check for manufacturing defects.

18. (Currently Amended) The computer readable medium as claimed in claim 17, wherein said FRIT kernel includes [[a]] the software built-in self-test engine (SBE) to execute the re-generative functional test of the complex device under test (DUT), and the expected test result obtained from computer modeling of the complex device under test (DUT) or from a known good device.

19. (Currently Amended) The computer readable medium as claimed in claim 18, wherein said software built-in self-test engine (SBE) of the FRIT kernel is programmed to generate and execute one or more ("N") instruction sequences, each sequence being executed on one or more (M) data sets where N and M represent an integer no less than "1" and are user-specified numbers used in generating the FRIT kernel ~~by an especially designed software tool~~.

20. (Original) The computer readable medium as claimed in claim 19, wherein said software built-in self-test engine (SBE) of the FRIT kernel is further programmed to generate one or more signatures to provide a unique identification of the test result of each test sequence and indicate whether the test result of a particular test sequence is "good" or "bad".

21. (Currently Amended) A test system for testing a complex device, comprising:

~~a complex device under test (DUT) having an on-board memory; and~~

~~a tester including~~ a tester memory; and

a controller to test a functionality of the complex device ~~under test (DUT)~~ by:

receiving [[the]] a functional random instruction test (FRIT) kernel [[in]] as kernel test patterns and storing the kernel test patterns in the tester memory;

loading the kernel test patterns stored in the tester memory onto [[the]] on-board memory of the complex device ~~under test (DUT)~~;

enabling execution[[, at]] of software within the FRIT kernel on the complex device ~~under test (DUT)~~, to perform a re-generative functional test of the complex device ~~under test (DUT) by applying the kernel test patterns to the complex device under test (DUT)~~; and

making a comparison between a test result of the re-generative functional test and an expected test result to check for manufacturing defects.

22. (Previously Presented) The test system as claimed in claim 21, wherein said FRIT kernel includes a software built-in self-test engine (SBE) to execute the re-generative functional test of the complex device under test (DUT), and the expected test result.

23. (Currently Amended) The test system as claimed in claim 22, wherein said expected test result is obtained from computer modeling of the complex device ~~under test (DUT)~~ or from a known good device.

24. (Currently Amended) The test system as claimed in claim 22, wherein said software built-in self-test engine (SBE) of the FRIT kernel comprises:

a RIT generator including compact RIT machine code in the on-board memory of the complex device ~~under test (DUT)~~ for generating the re-generated functional test;

a test program execution module including test execution directives for providing an environment to store and run the re-generated functional test; and

a test result compaction module including compression machine code to compress test results of the re-generated functional test for storage in the on-board memory of the complex device ~~under test (DUT)~~.

25. (Original) The test system as claimed in claim 24, wherein said test execution environment employs an exception handler for handling illegal conditions such as undesirable memory accesses, deadlock, shut-down, and infinite loops.

26. (Currently Amended) The test system as claimed in claim 21, wherein said complex device ~~under test (DUT)~~ includes a microprocessor.

27. (Currently Amended) The test system as claimed in claim 26, wherein, when ~~the kernel test patterns are applied to the microprocessor~~ software within the FRIT kernel is executed from the on-board memory, the microprocessor performs the following:

beginning a set-up for executing ~~the kernel test patterns~~ a software built-in self-test engine (SBE) within the FRIT kernel;

executing the ~~kernel test patterns~~ software built-in self-test engine (SBE) to generate a series of test sequences and associated data for respective test sequences;

running the test sequences, and at the end of the test sequences, obtaining the test results for storage in the on-board memory; and

dumping the test results ~~of the kernel test patterns~~ to the tester, via an interface, for making a comparison with the expected test result to check for manufacturing defects.

28. (Currently Amended) The test system as claimed in claim 27, wherein said FRIT kernel includes [[a]] the software built-in self-test engine (SBE) to execute the re-generative functional test of the complex device under test (DUT), and the expected test result obtained from computer modeling of the complex device under test (DUT) or from a known good device.

29. (Currently Amended) The test system as claimed in claim [[25]] 27, wherein said software built-in self-test engine (SBE) of the FRIT kernel is programmed to generate and execute one or more ("N") instruction sequences, each sequence being executed on one or more (M) data sets where N and M represent an integer no less than "1" and are user-specified numbers used in generating the FRIT kernel ~~by an especially designed software tool~~.

30. (Original) The test system as claimed in claim 29, wherein said software built-in self-test engine (SBE) of the FRIT kernel is further programmed to generate one or more

signatures to provide a unique identification of the test result of each test sequence and indicate whether the test result of a particular test sequence is "good" or "bad".

31. (Currently Amended) A complex device, comprising:

a memory to store a functional random instruction testing (FRIT) kernel [[in]] as kernel test patterns; and

a processor to perform a re-generative functional test of the complex device under test (DUT) upon execution of the FRIT kernel ~~test patterns~~ and to enable comparison between a test result of the re-generative functional test and an expected test result to check for manufacturing defects.

32. (Currently Amended) The complex device as claimed in claim 31, wherein said FRIT kernel includes a software built-in self-test engine (SBE) to execute the re-generative functional test of the complex device ~~under test (DUT)~~, and the expected test result.

33. (Currently Amended) The complex device as claimed in claim 32, wherein said expected test result is obtained from computer modeling of the complex device ~~under test (DUT)~~ or from a known good device.

34. (Currently Amended) The complex device as claimed in claim 32, wherein said software built-in self-test engine (SBE) of the FRIT kernel comprises:

a RIT generator including compact RIT machine code to reside in the ~~on-board~~ memory of the complex device ~~under test (DUT)~~ for generating the re-generated functional test;

a test program execution module including test execution directives for providing an environment to store and run the re-generated functional test; and

a test result compaction module including compression machine code to compress test results of the re-generated functional test for storage in the ~~on-board~~ memory of the complex device ~~under test (DUT)~~.

35. (Previously Presented) The complex device as claimed in claim 32, wherein said test execution environment employs an exception handler for handling illegal conditions such as undesirable memory accesses, deadlock, shut-down, and infinite loops.

36. (Currently Amended) The complex device as claimed in claim 32, wherein, when ~~the kernel test patterns are applied to the processor~~ software within the FRIT kernel is executed from the memory, the processor performs the following:

beginning a set-up for executing ~~the kernel test patterns~~ a software built-in self-test engine (SBE) within the FRIT kernel;

executing the ~~kernel test patterns~~ software built-in self-test engine (SBE) to generate a series of test sequences and associated data for respective test sequences;

running the test sequences, and at the end of the test sequences, obtaining the test results for storage in the on-board memory; and

dumping the test results ~~of the kernel test patterns~~ to the tester, via an interface, for enabling said comparison with the expected test result to check for manufacturing defects.

37. (Currently Amended) The complex device as claimed in claim 32, wherein said software built-in self-test engine (SBE) of the FRIT kernel is programmed to generate and execute one or more ("N") instruction sequences, each sequence being executed on one or more (M) data sets where N and M represent an integer no less then "1" and are user-specified numbers used in generating the FRIT kernel ~~by an especially designed software tool~~.

38. (Previously Presented) The complex device as claimed in claim 37, wherein said software built-in self-test engine (SBE) of the FRIT kernel is further programmed to generate one or more signatures to provide a unique identification of the test result of each test sequence and indicate whether the test result of a particular test sequence is "good" or "bad".